Push-Email And Mobile Devices In The Enterprise

Siemens AG, Corporate Technology
CT IC CERT
Dr. Heiko Patzlaff
heiko.patzlaff@siemens.com

Sophos PLC
Vanja Svajcer
vanja.svajcer@sophos.com

1. Introduction

Over the last couple of years smartphones have become an indispensible part of the IT infrastructure of enterprises worldwide. They are not only used for voice communications but people also use them to check their emails, schedule appointments, have access to the corporate directory, to store data and even to edit presentations and office documents.

There is no universal agreement on what constitutes a true smartphone but usually having a full feature set according to the above description is seen as an indication that separates a smartphone from the more common feature phones and basic phones.

The smartphone market, albeit still small compared to the overall phone market, has displayed strong growth over the last few years and is expected to continue to grow more dynamically then the rest of the industry. In 2007 about 120 million smartphones were sold worldwide, representing a 10% share of the overall phone market.

The security of smartphones is largely determined by the underlying operating system. The main contenders in this area are Symbian with a 65% market share, Windows Mobile based devices with 12%, Research in Motions BlackBerry with 11% and Apples iPhone with 7% worldwide market share. The market share in individual countries such as the United States differs substantially from these overall numbers.

The use of smartphones in enterprises puts some extra requirements on the security of these devices. While individuals demand a phone that is resistant to hacking attacks, worms and misuse in case it gets lost or stolen, enterprises also require easy administration, patching, enforcement of company policies, secure access to corporate resources etc. These requirements become more pressing the more capable mobile devices become with respect to storage space, processing power and connectivity.

A vital feature for the use of smartphones in the enterprise is push-email. The required infrastructure will be part of the considerations in the following chapters that compare the security of Symbian, WindowsMobile and BlackBerry devices.

2. History of Symbian, BlackBerry and Windows Mobile

Symbian is a multi-tasking capable microkernel operating system mainly used on ARM CPUs. It has its root in the EPOC operating system of the Psion PDA and is maintained and developed since 1998 by a consortium of vendors including Nokia, Motorola and Sony/Ericsson.
Symbian is the basis for several competing user interfaces, the two most significant being S60 and UIQ. By far the most widespread UI is the S60 interface used in Nokia phones. The UIQ interface is used by Sony/Ericsson and Motorola.

The current incarnation of the Symbian OS is version 9.5, released in 2007. Up to version 9.1 released in early 2005 the operating system provided only limited security features.
With version 9.1 a radical break was made which abandoned backward binary compatibility and introduced a new platform security model. Although most new phones running on Symbian use the S60 R3 interface which is based on Symbian 9.x, many older models in use are still running the S60 R2 software.

Research in Motion (RIM) introduced the first BlackBerry (BB) device in 1998. BlackBerries support PDA and mobile phone features but are most notable for their push-email functionality. The Push-Email feature of BlackBerry utilizes a proprietary protocol and requires a separate infrastructure component, the BlackBerry Enterprise Server (BES). Whereas BlackBerry traditionally appealed to business users, the current 8xxx model lineup include the BlackBerry Curve and the consumer oriented model BlackBerry Pearl that feature digital cameras and music players.

RIM licenses its email client to 3rd parties including Nokia which gives users the option to use a range of non-BlackBerry devices in a BlackBerry infrastructure.

Windows Mobile (WM) was originally introduced as the Pocket PC 2000 operating system in 2000. It is based on the Windows CE kernel and supports the Win32 API on mobile devices. The current version is Windows Mobile 6.1 based on Windows CE 5.0 but many smartphones in use still run on Windows Mobile 5.0. Microsoft introduced the DirectPush technology with its Messaging and Security Feature Pack (MSFT) in 2005. DirectPush can be deployed on an existing Exchange 2003 SP2 infrastructure and is supported by all new Windows Mobile based devices. Microsoft licenses DirectPush to 3rd parties and Nokia as well as Apple provide or will provide push-email support based on DirectPush on their devices.

Whereas Symbian and BlackBerry are tied to particular mobile device manufacturers Microsoft chooses to be device agnostic and license its operating system to a range of manufacturers. In the past the Taiwanese company HTC was the main provider of Windows Mobile based devices. Recently other larger companies such as Samsung, Sony/Ericsson and Motorola have licensed the operating system and provide handsets based on WM.

3. Push-Email Architectures and Risks

Whereas a standalone mobile phone poses a potential security problem only for the individual user the situation changes drastically when it is being used for accessing corporate resources.

The implementation of push-email requires the mobile device to become a network endpoint, constantly connected to and exchanging data with the corporate network. The compromise of a single device therefore impacts the security of the whole network. Moreover, with push-email, the mobile device exchanges and stores potentially sensitive data such as emails, appointments and contact data. How to secure this data on the device and in transit is another concern. Lastly, while an individual user might rightfully demand full control over the device - how he is allowed to use it, what applications he can install, which configuration and setup he is to choose - this freedom no longer is seen as a positive feature if the device is to be operated in a corporate environment. Since the user no longer is the data owner he needs to be restricted in the actions he can perform.

Various risks impact all three components of a push-email architecture - the mobile device, the transit network and the corporate network.

The main risks affecting the device are

- loss or theft
- loss of sensitive data
- malware
- unauthorized access (hacking)
- unauthorized modifications of security settings by the user
- loss of availability (spam)
- toll fraud (dialers)

While data is in transit it might potentially be intercepted, read, blocked or altered. Even if the actual content of the exchanged data is protected a third party might still be able to perform an analysis of the communication patterns that could reveal vital clues. If for example board members and other employees start exchanging email with a large outside investment firm this could be an indication of a pending carve-out of a troubled business unit.
This point is especially relevant if the third party has a holistic view of the traffic as is the case with mobile phone operators or governmental agencies in countries where regulations allow them access to this data.

Lastly, the mobile devices require access to the corporate network. Enabling this access might open up holes in the perimeter that can be attacked.

All mobile devices have in common that neither the end user nor the company deploying them has the same kind of control as is the case with other computing devices in the enterprise such as routers, laptops or printers. Whereas in the latter cases both hardware and software are provided by third party vendors at least some type of control remains in the sense that the data that they exchange can be monitored. This is no longer the case with smartphones. Here a large part of the hardware and software infrastructure in the network as well as in the device itself is controlled by the carrier.

4. Device Security Architectures

Although the difference between mobile devices and PCs becomes increasingly blurry, smartphones are still used differently from PCs and pose at least partly different security risks. Whereas on the PC platform different levels of trust are assigned to different users and this trust in turn extends to all applications a user is running, on the smartphone platform there is only one user. In order to still maintain a tired security model operating systems for mobile devices usually shift the focus from a user-centric model to a code-centric one. Identity is attached to code modules through code signing and evaluated at install or run-time in order to determine the trust assigned to the code.

How this code-centric security model is implemented and how effective it is varies between the different operating systems. Symbian and BlackBerry replace the multi-user with a multi-applications concept where each application has only access to its own data and the underlying operating system together with vital system settings provides a trusted computing base that is protected from tampering by applications and allows only well defined access via API functions. Microsoft more closely keeps with its traditional operating system concepts by using permission levels and security policies.

A large part of the security of a mobile device is application security. Application security answers questions like which applications are allowed to run, what they can do and if the user has a say in this. But besides this, there are also security decisions that are made when a user logs in or when a connection is established via Bluetooth or WAP. How powerful the support for these security decisions is, how fine grained and extensive the controls in these areas are, has a big impact on the security of a platform as a whole and the suitability for use in an enterprise. Lastly, the support a platform provides for the protection of sensitive data is important in a world where mobile devices get frequently stolen or lost and where user data is transmitted over open networks.

Symbian

Symbian, especially in its S60 R2 release, has for a long time been the main focus of hacking and malware attacks due to its widespread adoption and its comparatively weak security posture. This has fortunately changed with the radical break that came about with the release S60 R3. Due to the new binary format introduced in S60 R3, Malware written for the earlier release does not run anymore on later models.

S60 R3 and the Symbian OS version 9.x that underlies it use a new Platform Security Model that consists of three main components.
A trusted computing base (TCB) consisting of the core operating system kernel, drivers and config files, data caging that prevents an application from modifying the TCB as well as accessing code and data of other applications and capabilities which describe, on a per application basis, the operating system services an application is allowed to use.

Data caging restricts access by an application and therefore the user to only certain areas of the file system. Applications can access their own directories and certain directories that are marked as open. In particular access to the /sys directory containing all system binaries, certificates and settings is restricted to certified applications.

Capabilities divide the set of APIs into four classes. 60% of the API functions are open for all applications and don't require any permission checks. A second set of APIs involving access to user data require the user to consent once during the installation of the application.
A third set of APIs covering things like access to confidential user data, sensitive system data or the multimedia capabablities of the phone require the application to be signed through the Symbian Signed program which includes a functionality test.
Lastly a certain set of APIs covering DRM and access to the TCB is restricted to applications signed through the Manufacturer Approval program. Access to these APIs can be granted to 3rd party developers only by the device manufacturer.

Which capabilities an application needs is specified at compile time and stored in the application binary as an MMP file. At installation time this list is checked and the installer verifies if the application has been signed or certified an can be granted access to the requested capabilities. This model of compile time declaration and installation time checking

Arguably the most radical change came with the introduction of mandatory code signing in S60 R3 however. While the S60 R2 release was plagued with a rapidly growing malware problem, this has effectively stopped with release 3. While this clearly has a positive effect on the security of the platform some people rightfully argue that mandatory code signing takes away control from the user and puts it into the hands of Nokia. Similar to the restrictions on the iPhone which are used to tie the user to a particular service provider and which have been under constant attack by the community, the mandatory code signing restrictions on S60 R3 phones have recently been broken. These modifications however require a conscious act by the user and a temporary desktop connection and are therefore in their current form not a threat to the security of the platform.

The code signing procedure for the Symbian platform is rigorous and rather expensive. Application must follow strict guidelines in order to be signed. Normal applications are not allowed to hide from the task switcher for example. If access to a restricted API is required the developer has to state the reason for this.

BlackBerry

RIM acts as combined hardware and software provider. This business model allows it to exert strict control over the direction of the platform. Effective application security on BlackBerry devices is provided by a combination of the execution environment, code signing, IT policy rules and application control rules.

BlackBerries are made for use in the enterprise from the outset. They can be used as stand alone devices where the user is in control but the normal management model requires an administrator role that remotely manages and configures the device not only during provisioning but over the whole live cycle.

RIM uses various proprietary operating system kernels as a basis for its BlackBerry products. Applications are written in Java and run inside the Java 2.0 Micro Edition (J2ME) sandbox. They access the functionality of the platform via a well defined application programming interface (API) that consists of the standard J2ME API enhanced by additional classes provided by RIM. Traditionally this API was rather restricted. Lately RIM has broadened the API in order to better appeal to the consumer market.

A large part of the application security of the BlackBerry platform can be attributed to the security of the underlying Java subsystem. Practically all operations that have security implications are severely restricted. They either require the executable to be signed or the operation be confirmed by the user.

The BlackBerry platform parts with concepts familiar from traditional PC operating systems such as a file system or a registry and replaces them with more restricted mechanisms. Unsigned applications only have access to their own data store and can not influence each other. Furthermore, the security of the Java sandbox shields the OS and its system settings from access by 3rd party applications. Using Java as the execution environment on the other hand makes it difficult or impossible for third parties to provide security tools such as Anti-Virus scanners or firewalls.

3rd part applications on BlackBerries can be downloaded from the web, installed via a temporary desktop connection or pushed onto the device by the administrator. Which mechanisms are allowed is controlled by security policies.

RIM uses code signing to control access to RIM controlled API functions. Unsigned applications only have access to the limited functionality provided by the J2ME sandbox. This two level trust model is refined by application control rules.

Application control rules can be assigned on a per application basis and configure various rights an application has. They cover areas such as the ability to access the network, make phone calls or send SMS, use Bluetooth, access the keystore or send and receive email. The administrator can configure whitelists for individual applications by starting with restrictive default rules. Application control rules are somewhat similar to capabilities on the Symbian platform insofar as they provide fine grained control over what each application is allowed to do. But whereas Symbian puts the burden of the decision in the hands of the author on BlackBerry devices it is left to the administrator

The code signing procedure employed by RIM is only targeted at verifying the identity of a user. Like all such schemes it can be circumvented by using stolen credentials and credit card numbers.

An extensive set of about 300 IT policy rules allow the administrator to have comprehensive and fine grained control over the devices security.

Windows Mobile

Windows Mobile is based on Windows CE 5.0 as the operating system kernel. The operating system is made available by Microsoft and then modified and adjusted first by the hardware device vendor and then by the provider (branding). The operating system is modeled on Microsoft's PC operating systems. An extensive API is provided by the core OS. Most applications run native but Java and .NET frameworks are available. This model of a rich API similar to what is available on PCs is in marked contrast to both Symbian and BlackBerry which offer a limited feature set more specific to mobile devices that has expanded over time.

In order to solve the problem of trust in mobile devices Microsoft defines permission levels and roles and uses security policies stored in the registry to specify security settings. The permission model used by Microsoft is limited. There are only two levels, privileged and normal, where the privileged level allows unrestricted write access to system resources including the core operating system files and the registry. The normal permission level gives an application limited access to the file system, registry and API functions. All applications running with normal permissions have access to the data of other applications on the same level. The permission with which an application is allowed to run is determined by the certificate used for signing the application.

The problem of coarse grained permissions is compounded by the fact that the Windows Mobile operating system ships in two varieties. The so called one-tier devices, usually PDA phones without a keypad, only allow an all-or-nothing model. All applications run with privileged permission. The only control available to the user is the decision if an application should be allowed to run at all. The device can be configured to allow execution of all applications without further security checks, to allow only signed applications to run or to prompt the user before executing unsigned applications.

So called two-tier devices, usually phones with a keypad, differentiate between privileged and normal certificates and execute signed applications accordingly. Unsigned applications only run with normal privileges if permitted to run at all by the security policy.

Restricted APIs that can only be called by applications running with privileged rights cover areas such as memory access, debugging, driver loading, sending SMS or placing phone calls, functions that manipulate credentials or access the SIM card and others. On the file system and registry level, access to certain files and keys can be restricted by marking the entry with the system attribute.

In the past the code signing process for privileged and normal applications was a simple procedure that only required a developer to get a certificate from an approved certification authority such as VeriSign. In this process only the identity of the developer was ascertained. With Windows Mobile 6 Microsoft introduced an additional security check for privileged applications that require the developer to state the functionality of the application and submit it to a designated testing authority for verification.

In addition to permission levels Windows Mobile also defines and uses security roles such as SECROLE_MANAGER or SECROLE_USER_AUTH to control access to system resources. Security roles solve the problems that occur when a privileged process has to carry out an operation on behalf of another entity based on static content. If for example a configuration update is received over the air interface, the process that applies the new configuration settings to the system needs to run privileged in order to being able to write to restricted registry keys. To distinguish between trusted and less trusted configuration update providers a role is assigned to the configuration update file based on the origin of the update. When the configuration update is applied to the system and a restricted registry key needs to be modified, the role of the update XML file is checked against the role mask of the key. The latter is stored in the so called metabase.

The use of security roles gives OS vendors and operators more flexibility when dealing with multiple sources of configuration information.

The last component in the device security model of Windows Mobile are security policies. Security policies specify things like if applications stored on removable media cards should automatically be executed upon insertion, how to deal with unsigned applications, how the user needs to authenticate to the device, the strength of the PIN etc. These settings are stored in the registry and are either determined by the OEM, the operator or the user/enterprise administrator.

5. Sync Architectures and Central Administration

The last chapter dealt with the overall security architecture and application security in particular. They determine key features of the platform such as the questions if one application can access the data of another, if it can modify core operating system settings and components or if it has access to the system memory.

However, the effective security of a mobile device and the user data on it is also determined by things like data encryption, password strength, device locking, etc. For use in an enterprise it is desirable that these security features can be centrally and remotely controlled. Having central control over all mobile devices requires a sync architecture that pushes settings onto the devices in real-time. Once such a sync ecosystem is in place other things become possible. If a device gets stolen or lost a user can call the helpdesk which in turn issues a remote wipe request to the device that erases all sensitive data on it. Also email and software updates can be pushed onto the device as soon as they become available.

A sync architecture not only requires support on the mobile device but also network components inside the company. Both RIM and Microsoft provide complete sync architectures, but go different ways in implementing them. For Symbian based phones there are various offerings that provide this functionality such as the OMA based IntelliSync by Nokia - often licensing the corresponding technology from Microsoft or RIM.

The following discussion restricts itself to RIMs and Microsoft's sync architectures because of their widespread adoption.

Microsoft's DirectPush

Microsoft's DirectPush architecture was introduced in 2005 and is based on Exchange 2003 SP2 or later version. It works similar to Outlook Web Access (OWA). In it a mobile device establishes a secure HTTPS connection to an Exchange server and sends an update request with a large timeout. If no updates are available the connection will eventually time out. After the timeout the mobile device immediately opens an new HTTPS connection and send a new update request. When an update becomes available, say a new email has arrived or the central administrator has changed a policy setting, this update is send as a response to the update request over the HTTPS channel to the device which accepts it, closes the connections and reopens it again to wait for the next update.

In order to not drain the battery too much the timeout on the company firewall should be set to high values of 15-30 minutes. If the transit network times out the connection beforehand an algorithm is used by the device to find the longest possible timeout permitted by the transit network. These device initiated connections solve the problem that the IP address of the device is assigned dynamically and might change when a device loses connectivity or during handover to a different provider.

Besides the Exchange server which usually exists anyway no extra network components are required and not extra license fees accrue.
Of course this architecture also has its drawbacks. In order to not expose the Exchange server to the internet a firewall in the DMZ is required that acts as the HTTPS endpoint and authenticates the device. Authentication of the device via username and password opens up the possibility of denial of service attacks against the company user accounts since the firewall has no way of knowing if the incoming request is from a legitimate device or is a fake request from a rogue PC. Exchange 2007 introduced the possibility to also transmit and store the device ID which should allow two-factor authentication. Also, letting the firewall perform the authentication requires opening up various ports

from the DMZ to the active directory infrastructure inside the company network.

RIMs Sync Architecture

RIM uses a very different approach in solving the problem of changing endpoint IP addresses. Since the mobile operator assigns the IPs and is in control of the mapping between device ID and IP address, RIM engages with mobile operators worldwide for routing. Instead of using IP addresses it relies on the mobile operator to deliver a message to the correct phone. For this it uses its own worldwide infrastructure consisting of BlackBerry Enterprise Servers (BES) located in the company networks, distributed RIM data centers located on three continents and network connections from the data centers to the various mobile operators on one side and to the BES servers on the other.

When a mail arrives in a users inbox the BES is informed by the mail server and transmits the email via a proprietary protocol to the nearest RIM data center. Here the current provider of the handheld is determined and the data packet forwarded to it. The provider then makes sure that the packet is delivered to the right phone. For all transfers strong end-to-end encryption is used.

While BlackBerry is very popular in the US there has been concern particularly in Europe about the security of the data transfer.
For once some companies see the BES as a Trojan horse, connected to a Canadian company via a proprietary protocol and having full access to all email accounts of all BB users. The second concern is more serious however. The role of the data centers as concentrators puts RIM in a position where it has central control over the data traffic. If the end-to-end encryption can be broken or contains a backdoor from the outset, this would enable RIM to read all company emails that are not separately protected via PGP or S/Mime. Some intelligence agencies in Europe have hinted that Canadian and US agencies might be actively engaging with RIM for reading foreign email traffic. When RIM got permission to provide its services in China in 2006 after year long negotiations people were questioning how this fitted in with the Chinese obsession of controlling all communications. A similar discussion is going on right now with the Indian government and it remains to be seen how this plays out.

6. Central Administration and Patch Management

Both vendors use their sync architectures not only for push-email but also for device management and software provisioning.

BlackBerry administrators for some time enjoyed an extensive set of policy settings and all of them are available for remote configuration.
The amount of security settings that can be centrally administered by a Windows Mobile administrator has expanded with the introduction of Version 6 but is still severely limited. In its current form it does not cover much more than setting the password policy and the ability of remotely wiping the device. Microsoft has recently introduced its System Center Mobile Device Manager (SCMDM) solution to address these limitations. SCMDM only works with newer Windows Mobile devices however and so will take awhile before it gets a foothold in the enterprise.

One sore spot for all current mobile devices is patch management. Especially on the Windows Mobile platform there is no guaranty that security vulnerabilities can be patched and in the past Microsoft has provided frequent software updates but not individual security patches. This is only partly the fault of Microsoft although the clean separation of operating system components from modifications and additions by the handset vendor and mobile operator would certainly make it easier to roll out patches without risking the stability of the device.
Even if Microsoft made a patch available for a vulnerability, this patch had to be first approved by the device manufacturer and then made available by the mobile operator before it can land on the smartphone of the end user. In the past device manufacturers and mobile operators have reacted differently to updates provided by Microsoft.

7. Data Protection

Both Windows Mobile and BlackBerry offer device encryption natively. This encrypts all data stored on the internal flash memory when the device is locked. Only in their latest major revision of the OS has Microsoft extended this feature to also cover external memory cards. Problems remain however for the use in

enterprises since this encryption currently does not allow key escrow.
For Symbian based devices similar functionality is available with third party add-ons.

A problem of all mobile platforms is that RAM is battery buffered. Data in RAM therefore has to be seen more as static then as volatile. Although no public reports are known to the author where people were able to read the memory content of a locked device, it seems entirely feasible to break the security of lost or stolen smartphones by extracting credentials from the RAM. RIM at least seems to be aware of the problem and offers a security policy setting that enables automatic wiping of memory regions once they are freed during Java garbage collection.

Both Windows Mobile and BlackBerry platforms natively support S/Mime encryption. There is a problem in older Windows Mobile 5 devices that S/Mime encryption is only unacceptable 40bit strong. This has been corrected in WM 6 but should have been communicated by Microsoft more openly to its customers as a severe limitation.

All three platforms provide encryption capabilities and contain a secure storage for sensitive data such as passwords.
BlackBerries come with a firewall to provide additional perimeter security. This firewall governs not only IP based network access but extends also to the other communication features such as Bluetooth, SMS or MMS. Users and administrators can specify which incoming or outgoing connections are allowed based on type, source and destination.

8. Conclusion

Mobile devices pose specific security challenges. Stolen and lost devices raise the question of flash drive encryption, a constantly battery buffered RAM that of data protection in volatile memory.

Push-email architectures expose the secrets of a company to devices operating in a network of a third party mobile operator that not only has control over the infrastructure but also partly over the device itself and that has its own ideas about the security of its network that might contradict the requirements of the company.

All three vendors have significantly improved the security of their platforms over the last years. It can be argued that the close alignment of the Windows Mobile operating system family to the PC architecture makes it harder for Microsoft to achieve a similar level of security than the other vendors. But with the improvements that are about to come with the next version of Windows Mobile it seems unlikely that either of the three platforms will become mired in the same security problems that are with us on the desktop today.

9. References

1. iPhone bests Windows Mobile in U.S. sales (http://www.windowsfordevices.com/news/NS8948844785.html, accessed 03/25/2008)
2. Testing and Signing with Symbian Platform Security Version 1.3 (Symbian, March 2006)
3. Symbian Phone Security (Joob de Haas, BlackHat Europe, 2005)
4. Nokia Intellisync Device Management Functionality Guide (Nokia, 2007)
5. BlackBerry Enterprise Solution Security, Technical Overview, Version 4.1.0 (RIM, 2006)
6. BlackBerry Enterprise Server for Microsoft Exchange, Feature and Technical Overview, Version 4.1.2 (RIM, 2006)
7. BlackBerry Enterprise Server, Policy Reference Guide,Version 9 (RIM, 2007)
8. Protecting the BlackBerry device platform against malware, BlackBerry Enterprise Server Version 4.0 and later (RIM, 2006)
9. S/MIME Support Package Security, Technical Overview, Version 4.2 (RIM, 2006)
10. Security Concerns About NOC-Based Push E-Mail Architectures in Europe: Myth or Reality? (Gartner, August 2006)
11. The Pros and Cons of Using NOCs for Wireless E-Mail (Gartner, November 2005)
12. Whitepaper, Blackberry Security: Ripe for the Picking? (James O'Connor, Symantec Security Response, 2006)

13. Analyzing Complex Systems: The BlackBerry Case (FX of Phenoelit, BlackHat Europe 2006)
14. BlackBerry: Call to Arms, some provided (FtR & FX of Phenoelit)
15. Mobile Device Platforms, A Comparison of RIM Blackberry 4.0 and Microsoft Windows Mobile 5.0 Messaging and Security Feature Pack Enterprise Mobile Solutions (Wipro Technologies, October 2005)
16. Mobile Messaging with Microsoft Exchange Server 2003 Service Pack 2 and Windows Mobile 5.0 Messaging and Security Feature Pack (Microsoft, November 2005)
17. Security of Smart Phones, Master Thesis (Collin Richard Mulliner, University of California Santa Barbara, June 2006)
18. Advanced Attacks Against PocketPC Phones (Collin Mulliner, 23rd Chaos Communication Congress, December 2006)
19. Secure Windows Mobile Development and Deployment (Chung Webster, Microsoft, August 2004)
20. Step-by-Step Guide to Deploying Microsoft Exchange Server 2003 SP2 Mobile Messaging with Windows Mobile 5.0-based Devices (Microsoft, March 2006)
21. Network Security for the Windows Mobile Software Platform, Whitepaper (Microsoft, March 2006)
22. Virusability of Modern Mobile Environments (Vesselin Bontchev, VB Conference 2007)